John T. Anderson

Engineering Note

Date: July 18th, 2002
Rev Date: July 18th, 2002

Project: Central Preshower Detector
Doc. No: A1020718

Subject: Discriminator data path for CPS fibers in AFE boards

Introduction

The Analog Front End board (AFE), as is well known, takes input from the Fiber Tracker fibers and produces one bit per fiber (discriminator data) that is sent over LVDS gigabit links to the Digital Front End (DFE) boards through the Mixer subsystem. The path for the CFT Axial fibers is relatively simple in that every block of 64 fibers passes through a single multi-chip-module (MCM), with the data from each of the eight MCMs handled independently.

This is not true in the Central Preshower (CPS) section of the Fiber Tracker. The CPS Axial fibers are cut in to North and South halves with each half-fiber being connected to a unique AFE channel. While this provides pleasant differentiation for some tracking purposes, for the main trigger the North and South halves must be electrically recombined to provide a single discriminator bit which is set if *either* half of the fiber is "hit". While this might seem a job better suited to the large FPGAs found within the DFE and the Mixer, the designs of these boards and the cabling system are such that this excess task is imposed upon the AFE. Fortunately, the AFE contains "share" buses between the LVDS_MUX CPLDs which allow the hardware to provide the additional functionality without adding any wires to the board.

CPS discriminator bits cannot be merged carelessly. The Mixer requires that the bits be presented in a very particular order. In the original system design (when a 12-MCM AFE was still in the works) a 28-bit LVDS link[1] was specified to carry the CPS Axial data. Since the 12-MCM AFE was never manufactured, additional 8-MCM AFE boards, which only have 21-bit LVDS links, have been conscripted to take their place. This introduces complex routing and timing requirements which shall be detailed later in this note. The remainder of this document shall then show how the extra routing resources designed into the AFE, plus special code in multiple PLDs, allow the preshower bits to be correctly merged and propagated.

LVDS Link Patterns from AFE to Mixer

In order to derive the necessary bit pattern, one must go backwards from the Mixer to the AFE. The Mixer, designed with the expectation of receiving data from the 12-MCM AFE, expects to receive 'high threshold' CPS data and 'low threshold' CPS data – two output bits for every CPS input channel of the AFE. This is because the 12-MCM AFE was to implement two MCMs for each CPS input, with charge splitting that would provide the dual threshold. The 8-MCM AFEs actually used in the detector only have one output bit per input channel, such that only *half* of the Mixer data is actually populated. Flexibility in the Mixer and the DFE allow us to map the 8-MCM AFE's *single* CPS output bit onto *either* the 'high threshold' *or* the 'low threshold' bit of the Mixer, providing us with a way to escape the 28-bit/21-bit issue alluded to earlier.

[1] LVDS links are specified as being "21-bit" or "28-bit" by convention. More precisely, a "21-bit" link sends seven 21-bit packets (147 bits) every 132 nsec, and a "28-bit" link sends seven 28-bit packets (196 bits) every 132 nsec. A "28-bit" receiver can receive and understand data sent by a "21-bit" transmitter, albeit with "holes" in each 28-bit pattern received. Certain bits of the "28-bit" receiver's pattern can never be driven if a "21-bit" transmitter is used.

28-bit vs. 21-bit link map

Table 1 shows how the 28-bit receiver of the Mixer 'sees' the 21-bit data sent by the 8-MCM AFE. When a 21-bit driver is connected to a 28-bit receiver, bits 27,23,17,16,11,10 and 5 cannot be used at the receiving end because those bits are not driven. By convention, data can also not be sent on bit 0 because that bit is reserved as the frame marker. One of every seven time-slices, the frame marker is set to '1' indicating not only 'end of frame' but also passing along the crossing clock frequency.

Sent		20	19	18		17	16	15	14	13			12	11	10	09			08	07	06	05		04	03	02	01	00
Rcvd	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Slice 1																												0
Slice 2																												0
Slice 3																												0
Slice 4																												0
Slice 5																												0
Slice 6																												0
Slice 7																												1

Table 1

CPS Data at the Mixer

The Mixer receives the data from the AFE and it re-maps the data into the sector organization required by the DFE. The link from Mixer to DFE introduces the concept of color-coding the bits. Table 2 shows the same picture as Table 1, but with colors. The colors indicate groups of bits that end up on the same wires to the DFE. The open boxes indicate unused bits, and the grey boxes indicate where extra status information is packed. It will be seen that some of the colored boxes are in positions which Table 1 declares unusable, indicated by the heavy border around these bits. For these bits, then, the position is fixed – the other bit of the high/low pair must contain the information. For all the other bits, however, we are free to send the one bit of AFE information on *either* the high bit or the low bit, as shown in the Mixer documentation.

All of the 'high threshold' bits have an 'H' in Table 2 and all of the 'low threshold' bits have an 'L' in Table 2. Individual bit numberings within the Mixer have been eliminated for ease of reading. For every 'H' bit of a given color there is a matching 'L' bit with the same coloration. *Only one of them need be driven*. We're free to choose which one we want to drive, so there are lots of choices. The obvious first reduction is to eliminate those bits which fall in the undrivable positions, since their mate is the one that *must* be used. For instance, seven of the eight solid-purple 'H' bits are unusable, so that means that the seven corresponding solid-purple 'L' bits are required.

A couple of the extra status bits fall where we can't drive them. This has been handled by some recoding in the Mixer since the original design.

Sent		20	19	18		17	16	15	14	13			12	11	10	09			08	07	06	05		04	03	02	01	00
Rcvd	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Slice 1	H	H	H	H	H	H		H	H	H	H		H		L		L	L		L		L	L	L	L		L	0
Slice 2	H	H	H	H		H		H		H	H	H	H		L		L	L		L		L		L	L		L	0
Slice 3		H	H	H		H		H	H	H	H		H	L	L		L	L		L		L		L	L		L	0
Slice 4		H	H	H		H	H	H		H	H		H		L		L	L		L		L	L	L	L	L	L	0
Slice 5		H	H	H		H		H		H	H		H		L		L	L		L		L		L	L		L	0
Slice 6	H	H	H	H		H		H		H	H		H		L	L	L	L		L	L	L		L	L		L	0
Slice 7		H	H	H		H		H		H	H		H		L	L	L	L	L	L		L		L	L		L	1

Table 2

Internal AFE Sharing of Bits

As noted in the Introduction, the AFE has the job of taking the two CPS bits from the North and South halves of the CPS strip and performing the logical OR function necessary to recombine the strips. Further, the internal AFE architecture is such that the 20 bits of CPS data (bits 20..01) are sent from two separate chips – one associated with each MCM, and with each half of the LVDS link. One chip drives bits 20..11 and the other drives bits 10..01. To perform the OR function, each of the two chips has to not only send data from its own MCM to the LVDS driver, it must also send some of its local bits over to the adjacent chip, and also receive some of the adjacent chip's bits to OR with those local bits not sent to the adjacent one. There are only 10 wires connecting the two chips, so the sum of bits passed can be no more than 10 in any timeslice. This will impose some further restrictions on the choices which can be made.

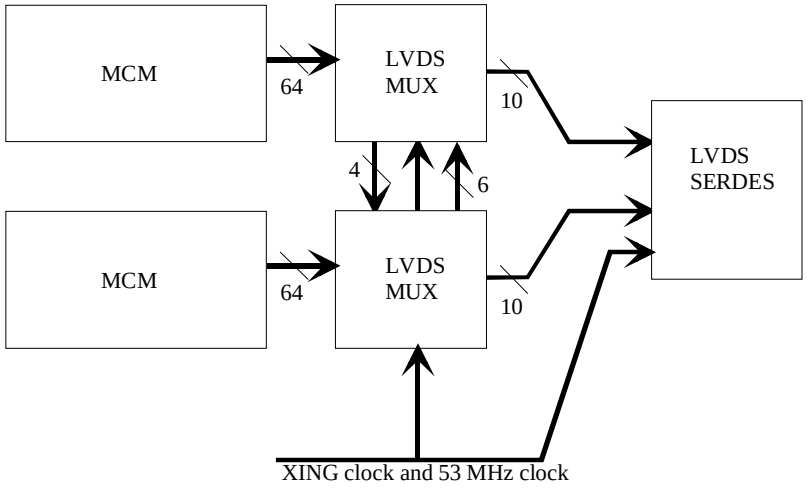

Figure 1

Mixer & DFE Restrictions

Various bits must leave the Mixer on certain time-slices in the data transfer between Mixer and DFE. This doesn't affect the AFE, so is not treated here, but bear in mind that the methodology described is glossing over the fact the Mixer firmware and DFE firmware have yet more timing restrictions which must be addressed.

The Final Map

Gory details removed, Table 3 shows how the solution works out. Jamieson has reduced Table 2 so that only the 'H' or 'L' bit of each pair is driven. The original table violated the sharing total restriction, and so we shuffled again to obtain this final version. All 64 CPS bits are driven on *either* the original CPSn 'H' or 'L' position, except for three. The CPS30, CPS31 and CPS32 bits have to be moved from their original positions to resolve the AFE share limitation. This is handled by a slight recoding in the DFE firmware as follows:

- CPS30 was originally in bit 20, slice 6, and is moved to bit 20, slice 5. This is permissible because bit 20, slice 5 was also a 'solid red' bit (CPS29), and requires no recoding in the Mixer.
- CPS31 was originally in bit 20, slice 7, and is moved to bit 20, slice 6, freed up by the move of CPS30.
- CPS32 was origianlly in bits 21, slice 4, and is moved to bit 20, slice 7, freed up by the move of CPS31.

Table 3 has been reformetted to show only the 21 bits actually driven, so bits 27,23,17,16,11,10 and 5 of the 28-bit link are removed. This allows a more clear view of the data *as it is on the AFE*. The table is separated into three sections indicative of the 'high' half of the link (driven from one PLD), the 'low' half of the link (driven from a different PLD) and the 'sync' bit (generated by neither PLD, but by global clock circuitry). Two new columns are added that show the number of bits which must pass from 'high' to 'low', and from 'low' to 'high', to implement the OR logic. A moment's inspection shows that the sum of these two columns is the total number of bit sharing wires required, and this sum never exceeds 10, so this solution can be implemented in the hardware (as discussed above in the Internal sharing of AFE Bits section).

	20	19	18	17	16	15	14	13	12	11	# bits		# bits	10	9	8	7	6	5	4	3	2	1		0
	26	25	24	22	21	20	19	18	15	14	-→		←-	13	12	9	8	7	6	4	3	2	1		0
1	CPS 62	CPS 51	CPS 44	0	0	0	0	0	CPS 6	0	6		4	CPS 55	0	0	CPS 33	0	CPS 23	CPS 17	CPS 11	0	CPS 1		0
2	0	0	CPS 45	CPS 41	0	0	0	CPS 21	CPS 7	0	5		4	CPS 56	0	0	CPS 34	0	CPS 24	0	CPS 12	0	CPS 2		0
3	CPS 63	CPS 52	CPS 46	0	0	0	0	0	CPS 8	0	5		4	CPS 57	0	0	CPS 35	0	CPS 25	0	CPS 13	0	CPS 3		0
4	0	CPS 53	CPS 47	CPS 43	0	0	0	0	CPS 9	0	6		4	CPS 58	0	0	CPS 36	0	CPS 26	CPS 18	CPS 14	0	CPS 4		0
5	CPS 64	0	CPS 48	0	0	CPS 30	0	0	CPS 10	0	5		4	CPS 59	0	0	CPS 37	0	CPS 27	CPS 19	CPS 15	0	0		0
6	0	0	CPS 49	CPS 40	0	CPS 31	0	CPS 22	0	0	5		4	CPS 60	0	0	CPS 38	0	CPS 28	0	CPS 16	0	CPS 5		0
7	0	CPS 54	CPS 50	CPS 42	0	CPS 32	0	0	0	0	4		4	CPS 61	0	0	CPS 39	0	CPS 29	CPS 20	0	0	0		1

Table 3

Physical Implementation of Map in the AFE

Each of the two LVDS_MUX PLDs in a pair are reprogrammed to have three major blocks:

1. A block of bits from the MCM that are passed to the neighbor PLD;
2. A block of bits received from the neighbor PLD;
3. A block of bits from the MCM that are ORed with the bits of (2) to form the actual output to the LVDS driver.

The CLOCKGEN is modified from the original version such that the LD_SHFT signal, which marks the time when the discriminator bits are to be loaded, is issued one crossing earlier than in the regular CFT boards. This signal is distributed on a single wire to every LVDS_MUX CPLD on the board. Normally, LD_SHFT is set to occur about 50 nsec after the S/H clock. Most of the 50 nsec are to allow the output of the SIFT chip to stabilize, because the SIFT discriminator has terrible time walk for signals near the threshold. In the CPS, the LD_SHFT is sent early by 132 nsec, but a delay of six ticks of the 53 MHz clock is added in each LVDS_MUX CPLD such that the effective time of the LD_SHFT for CPS data streams is 18.9 nsec *early* relative to the CFT. Scope measurements at the test bench indicate that this one clock tick can be gained without badly affecting the threshold stability, but no more.

The CFT Stereo channels of a CPS/CFT Stereo AFE also receive the earlier LD_SHFT, but their LVDS_MUX CPLDs implement a full seven ticks of additional delay such that the CFT Stereo data is timed identically to that on the regular all-CFT boards. The extra clock tick gained by moving the LD_SHFT earlier is used by the CPS LVDS_MUX PLDs to perform the sharing of data bits between them.

The two LVDS_MUX CPLDs each connect to all 64 bits of discriminator data from their local MCM. The CPLD that is connected to the lower 10 bits of the LVDS link uses 4 of the 10 bits of interconnection between the two CPLDs to send data to the 'upper' CPLD. The other six bits are used by the 'upper' to send bits to the 'lower'. When the LD_SHFT signal arrives at the two CPLDs they both begin the multiplex process. *The data from the MCM is assumed to be latched by the MCM (via the S/H signal) such that the MCM presents stable data throughout the multiplexing process. No holding register is implemented in the LVDS_MUX*. On the first tick of the clock, both CPLDs will capture the bits associated with CPS fibers 1,6,11,17,23,33,44,51,55 and 62 of the 64 presented by each MCM. That's right, *both* PLDs capture *all* the bits in the row of Table 3 above.

The 'lower' PLD sends bits 6, 44, 51 and 62 over to the 'upper' PLD, where on the next tick of the clock these four bits will be combined with bits 6, 44, 51 and 62 of the 'upper' MCM to form the four OR bits. At the same time, the 'upper' PLD is sending bits 1, 11, 17, 23, 33 and 55 over to the 'lower', where on the same clock tick the 'lower' PLD ORs them with its local MCM bits on the way out. While the OR functions of the bits on line 1 of Table 3 are being registered the next bits are pipelined. This is probably best described in a table; in Table 4, Clock Tick 1 is when the LD_SHFT edge is sampled by the master clock.

Clock Tick	**Upper OR out**	**Upper local bits**	**Upper OR in**	**Upper LVDS out**	**Lower OR out**	**Lower local bits**	**Lower OR in**	**Lower LVDS out**
1	1,11,17,23,33, 55	N/A	N/A	N/A	6,44,51,62	N/A	N/A	N/A
2	2,12,24,34,56	6,44,51,62	6,44,51,62	N/A	7,21,41,45	1,11,17,23,33, 55	1,11,17,23,33, 55	N/A
3	3,13,25,35,57	7,21,41,45	7,21,41,45	Bit-by-bit OR of 6,44,51,62	8,46,52,63	2,12,24,34,56	2,12,24,34,56	Bit-by-bit OR of 1,11,17,23,33, 55
4	4,14,18,26,36, 58	8,46,52,63	8,46,52,63	Bit-by-bit OR of 7,21,41,45	9,43,47,53	3,13,25,35,57	3,13,25,35,57	Bit-by-bit OR of 2,12,24,34,56
5	15,19,27, 37,59	9,43,47,53	9,43,47,53	Bit-by-bit OR of 8,46,52,63	10,30,48, 64	4,14,18,26,36, 58	4,14,18,26,36, 58	Bit-by-bit OR of 3,13,25,35,57
6	5,16,26,38,60	10,30,48, 64	10,30,48, 64	Bit-by-bit OR of 9,43,47,53	22,31,40, 49	15,19,27, 37,59	15,19,27, 37,59	Bit-by-bit OR of 4,14,18,26,36, 58
7	20,29,39, 61	22,31,40, 49	22,31,40, 49	Bit-by-bit OR of 10,30,48, 64	32,42,50, 54	5,16,26,38,60	5,16,26,38,60	Bit-by-bit OR of 15,19,27, 37,59
8	N/A	32,42,50, 54	32,42,50, 54	Bit-by-bit OR of 22,31,40, 49	N/A	20,29,39, 61	20,29,39, 61	Bit-by-bit OR of 5,16,26,38,60
9	N/A	N/A	N/A	Bit-by-bit OR of 32,42,50, 54	N/A	N/A	N/A	Bit-by-bit OR of 20,29,39, 61

Table 4

Timing Considerations in Detail

Note that it takes 9 ticks of the 53 MHz clock to get all the data out, whereas the beam crossing occurs every seven ticks of the clock. At first glance it would appear that this algorithm cannot work when the accelerator starts sending beam every 132 nsec. However, the LVDS_MUX PLDs have the required two layers of pipeline in them. The first layer of pipeline is located in the local multiplex logic and also in the shared line inputs, which are both registered. The outputs to the LVDS drivers are again registered, creating the second pipeline layer. Thus, if new data were presented to the inputs starting at clock tick 8 in table 4 above, nothing is lost, so long as the new data assertion occurs sufficiently after tick 7 to insure the propagation to the adjacent PLD and sufficiently before tick 8 to guarantee setup times; roughy, the data transition window needs to be less than about 10 nsec. The SIFT chip is nowhere near fast enough to do this, so addition of more flip-flops is likely a necessity.

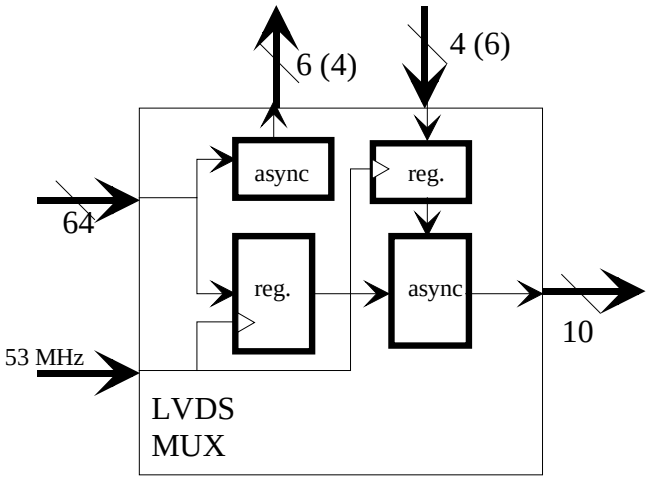

Figure 2

The propagation delay through the asynchronous selection block and over the board to the adjacent LVDS_MUX CPLD is significantly less than the 18.9 nsec period of the 53 MHz clock and so no setup or hold violation will occur. Similarly, the propagation through the asynchronous combining block of each LVDS_MUX plus the delay from LVDS_MUX to LVDS SERDES is also well less than 18.9 nsec, so again, setup times should not be violated. Scope measurements are required in order to insure that things are OK; they should be quite similar to the current AFE program load which gives about 7 nsec of setup and 11 nsec of hold time for LVDS data at the input of the SERDES.

Other Timing Tweaks

The 53 MHz clock of the AFE board is generated using an eight-output PLL. The outputs come in pairs and a skew can be introduced between pairs. The clock skew control inputs are ‘three-state’, meaning that 0V, 2.5V and 5V are considered three distinct logic levels. Two skew control bits are provided per output pair, so the three states yield eight skew possibilities. Connection to the skew control bits made through the CLOCKGEN CPLD. With the normal CLOCKGEN load the skew bits are fixed and cannot be changed by the user. If data errors are seen in the CPS implementation adjustment of the LVDS_MUX 53 MHz clock may be made relative to the SERDES 53 MHz clock by redesigning the CLOCKGEN CPLD. The skew controls can adjust the setup time a few nanoseconds in either direction. This is not expected to be required.